Votre site web est obè

21 . 05 . 2014

Certains d'entre vous ont sûrement connu l'époque du 56k. En ce temps là, charger une se
demander plusieurs minutes. Puis est arrivé l'ADSL et le haut débit qui ont démultiplié la
j'ai l'impression que nous passons toujours autant de temps à attendre le chargement d'u

De nos jours les sites sont de plus en plus complexes, embarquent de plus en plus de contenu.
se chargent de plus en plus vite… En théorie car paradoxalement le web évolue vite, mais les si
charger.

Il existe beaucoup d'options pour améliorer la performance de votre site internet. Pour vous aid
place ces solutions j'ai découpé cet article en 3 parties :

- Geek
- Guru
- Hacker

Chaque technique peut être déployée partiellement, en partie ou pas du tout. Chaque palier pe
de votre site suivant son niveau de complexité.

Geek de la performance

Les 2 choses les plus simples à mettre en place rapidement sont la compression des fichiers et l
indispensables vous aiderons à :

- Réduire le poids des pages.
- Réduire le nombre de requêtes http, et donc la latence.

Puisque ces modifications ont lieu côté serveur, elles amélioreront immédiatement les performa

ACTIVER `MOD_DEFLATE` POUR LA COMPRESSION `GZIP` DE VOS FIC

De la même manière que nous zippons nos pièces jointes envoyées par email, votre serveur pe
les envoyer au navigateur. Pour **préserver la bande passante** de vos utilisateurs mobiles, il
la taille de votre site.

La librairie de compression pour Apache `mod_deflate` fonctionne uniquement sur des fichiers
`JavaScript`. Généralement les images `.jpg` sont déjà compressées, cette librairie n'ont donc a

La mise en place demande 3 étapes :

1. Aller sur le projet **GitHub HTML5Boilerplate**.
2. Copier le contenu `mod_deflate` du fichier `.htaccess`,
3. et le coller sur votre serveur.

METTRE LES RESSOURCES EN CACHE

Si votre site est correctement architecturé, les mises à jours des ressources `CSS`, `JavaScript` e
le contenu. Dans un contexte mobile nous pouvons télécharger ces ressources statiques en une
l'appareil.

- Moins de requêtes http seront effectuées.
- Le contenu total à télécharger est diminué.

De cette façon, pour les futures visites, l'expérience de navigation sera incroyablement plus rap
consiste à utiliser les entêtes d'expiration avec une date adaptée.

Il n'est pas conseillé de spécifier une durée inférieure à 1 mois et supérieure à un an. Si vous c
vous pouvez :

- Spécifier et identifier cette ressource dans votre fichier `.htaccess`.
- Modifier le nom du fichier, qui est une bien meilleure solution car cela donnera naissance à u

La mise en place demande 3 étapes :

1. Aller sur le projet **GitHub HTML5Boilerplate**.
2. Copier le fichier `.htaccess`, pour la partie `mod_expires`,
3. et le coller sur votre serveur.

La plus grande limite de cette technique est la taille du cache des appareils mobiles. Un site we
cache des appareils Android 2.x est limité à 6Mo, les appareils iOS sont plus avantagés avec un

Pour contourner ce problème nous pouvons mettre en place une fonctionnalité en utilisant l'AP
cela il faut être guru de la performance.

Hacker de la performance

La compression et la mise en cache des fichiers sont de bonnes pratiques mais ce n'est que la b
la performance et offrir une expérience optimisée à vos utilisateurs vous allez devoir travaille
permettent de :

- Réduire le poids de vos pages.
- Réduire le nombre de requêtes à votre serveur, et donc la latence.
- Afficher du contenu à la demande.

Les techniques de minification et de concaténation devraient êtres mises en places lors de déplo
test / production. Elles ne sont pas vraiment utiles si vos mises à jours sont fréquentes et multi

METTRE VOS FICHIERS AU RÉGIME.

Minifier votre code source signifie simplement supprimer les octets inutiles d'un fichier (espaces
syntaxe est indispensable pour qu'un humain comprenne, mais elle n'est pas nécessaire pour l'é

> *Le code source est écrit pour les humain, pas pour les machines*
>
> *— @drublic*

Avec la minification le navigateur télécharge moins de contenu, le rendu de votre page est don

Si vous utilisez une librairie populaire comme jQuery vous utilisez certainement la version `.mir`
disponible pour le JavaScript, mais peut être appliquée à n'importe quelle type de ressource tex
minifiée dans un code de production est le CSS. La plus vieille solution pour réaliser cette opér
yahoo! Si vous êtes un **développeur front-end éclairé** essayez de chercher du côté des tâc
opération.

LA CONCATÉNATION : UN FICHIER POUR LES GOUVERNER TOU

Si le père de la performance est la minification, la concaténation des fichiers est sans doute la n
télécharger 10 fichiers JavaScripts qui demandent 10 requêtes http au serveur, les utilisateurs d
fichiers qui représentent la combinaison de ces 10 fichiers.

Après les `mod_expires`, la concaténation est la meilleure option à mettre en place pour fluidifi
Malheureusement la pratique est plus complexe que la théorie.

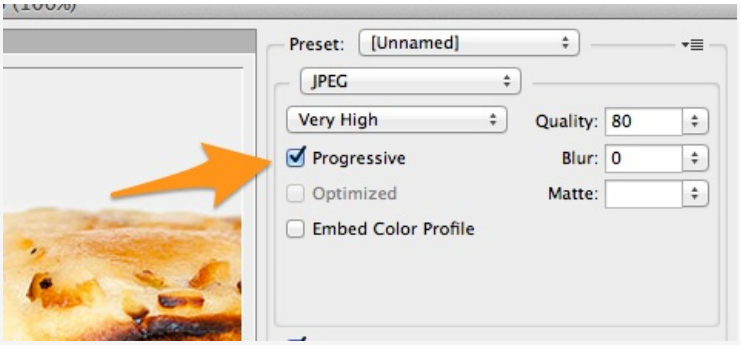
En moyenne un navigateur peut télécharger 6 fichiers en parallèle. Comme la minification YUI e
fichiers. Dans un environement ruby c'est l'`asset pipeline` qui se charge du boulot, en php c
aussi **construire votre propre asset pipeline avec gulp**

OPTIMISATION DES IMAGES

Les images sont les éléments qui prennent le plus de place. En moyenne, c'est plus de 60% da
chacun, développeur comme designer, d'optimiser le poids des images. Le processus se fait en

1. Choisir le format adapté

`.png` pour les aplats de couleurs, `.jpg` pour les photographies. N'enregistrez **jamais** vos imag
essayez de trouver le bon compromis entre qualité / poids.



Avec Photoshop `fichier` > `enregistrer pour le web`

N'oubliez pas de cocher la case **jpg progressif**, cette technique utilisée pour les débits lents à
sens aujourd'hui dans un **contexte de navigation mobile**.

2. Supprimer les informations inutiles

Pour gagner quelques octets supplémentaires il est possible de supprimer d'autres informations
(métadonnées, informations de claques, ...) Cette opération peut se faire très facilement avec d

- **imgOptim** pour macOS
- **pngGauntlet** pour windows

Si vous souhaitez en savoir plus sur l'optimisation des images, je vous invite chaudement à lire
sur l'optimisation intelligente des images `.png` et `.jpg`

- **Optimiser ses images png**
- **Optimiser ses fichiers jpg**

3. Automatiser la tâche

Avec la puissance de node et **ImageOptim-CLI** il est possible d'automatiser tout ce processu
temps considérable.

```
npm install -g imageoptim-cli
```

REPORTER LE CHARGEMENT DU CONTENU

Toutes les techniques précédentes, ormis la concaténation, vous aiderons à réduire la taille de v
prioriser le chargement de votre contenu.

La performance n'augmentera pas, c'est le sentiment de vitesse que ressentira l'utilisateur qui d
simple de retarder le chargement d'un fichier JavaScript en utilisant l'attribut HTML5 `defer` ou

```
<script defer src="application.js"></script>
<script async src="application.js"></script>
```

D'un point de vue latence et performance ces attributs se comportent de façon similaire. Ils den
scripts seulement après avoir téléchargé l'intégralité du contenu de la page.

Vous devez vous poser la question : quel priorité dois-je utiliser ?

`defer` charge les scripts dans l'ordre d'appel

Supposons que vous chargez la librairie jQuery ainsi qu'un autre fichier de plug-in. Ces 2 fichie
script doit être chargé *après* le premier.

`async` le chargement de l'ordre des fichiers n'a aucune importance

Guru de la performance

Les geeks et les hackers de la performance utilisent des techniques d'optimisation traditionnelle
internet, bien avant l'apparition du responsive design.

Les techniques de guru peuvent êtres mises en place pour des applications web HTML5 avec ur
Elles permettent de :

- Réduire le poids de vos pages.
- Réduire le nombre de requêtes http, et donc la latence.
- Optimiser le chargement du contenu pour être utilisé plus rapidement.

Le principal désavantage est le support des navigateurs

Dans certains cas il ne sera pas possible de garantir le résultat sur de vieux navigateurs. N'hésit
fonctionnalités sur le site **caniuse**.

ORDRE DES SOURCES

Curieusement organiser correctement votre CSS et vos fichiers JavaScript peut avoir un large in
chargement.

> *JavaScript bloque le chargement de votre contenu*
>
> — ***Manifeste d'un développeur front-end***

Voici un très mauvais exemple que je rencontre quotidiennement :

```
<head>
  <script src="jquery.js"></script>
  <link rel="stylesheet" href="small.css">
  <link rel="stylesheet" href="big.css">
</head>
...
```

Vous ne devez **JAMAIS** faire cela. Le JavaScript bloque téléchargement des autres ressources e
exception votre JavaScript doit toujours être appelé uniquement avant la fin du `</body>`

`LOCALSTORAGE` COMME CACHE NAVIGATEUR

`localStorage` est une spécification du W3C. Cette technologie est comparable aux cookies uti
stocker des informations de type clef / valeur. Elles sont sauvegardées même si le navigateur es
les informations associées au `localStorage` ne sont jamais renvoyées au serveur depuis le cli

Voici un exemple assez basique : utiliser `localStorage` pour mettre en cache jQuery

1. Vérifier si le fichier existe
2. Charger le contenu en AJAX
3. Insérer le contenu dans la page avec `js-jquery`

Il faut noter que `localStorage` ne fonctionne qu'avec des `string`. Si vous souhaiter utiliser c
images, vous devez les parser en base64 d'abord.

`localStorage` est une solution assez facile à mettre en place pour des choses basiques. Si vou
conseille d'essayer **basket.js** maintenu par Addy Osmani qui permet de prioriser le chargeme

LAZY LOADING JAVASCRIPT

Si votre application web / site internet embarque beaucoups de JavaScript il peut être intéressa
loading pour réduire le temps de démarrage.

Quand un navigateur télécharge du JavaScript, il le **parcours et l'exécute**. Comme expliqué
de la page jusqu'à ce que le processus soit terminé. Dans la plupart des cas, vous n'avez sûrem
JavaScript soit parcouru et exécuté au chargement de la page. Cela porte préjudice à la perforn

Il existe 2 techniques pour corriger le problème :

1. Modifier le script précédemment utilisé pour `localStorage` en utilisant JavaScript pour inse
2. Utiliser `XmlHttpRequest` pour **télécharger et évaluer** le code. Le principal problème est
   une fonctionnalité qui n'est pas encore téléchargé cela risque de "casser" la mise en page.

L'équipe qui s'occupe de Gmail a trouvé une solution pour que le code ne soit pas exécuté : **co**

Le navigateur télécharge et évalue l'intégralité de votre JavaScript, mais comme ce dernier est c
méthode libère du CPU pour effectuer d'autres opérations. Quand l'utilisateur a besoin d'une fo
décommentée et évaluée. Voici un exemple :

Cette technique est complexe à mettre en place car votre architecture JavaScript doit être irrépr

LE CAS DES BOUTONS SOCIAUX

Fin 2011, le magazine Allemand Heise a publié un nouvel ensemble d'**icônes pour les réseau**
? Il faut 2 clics pour "aimer", "tweeter" ou ajouter "+1".

De base les images des icones sont chargées. Si l'utilisateur souhaite partager, alors les widgets
solution présente de nombreux avantages non négligeables.

1. Les utilisateurs ne sont pas constamment trackés. Les widgets ne sont activés que si l'utilisat
2. Extrèmement efficace avec du responsive design.

Zurb

> "Charger les boutons so
> Google+, Facebook et T
> totalise 19 requêtes se
> et prend 246.7ko en b
> passante.

Pourquoi les boutons sociaux tuent votre site web ?

De plus vous ne téléchargez que ce qui est nécessaire. Si un utilisateur n'est actif que sur twitte
facebook n'a pas de sens, et cela lui évite un temps de latence en supprimant les requêtes http

UTILISER DES SPRITES POUR CONCATÉNER VOS IMAGES

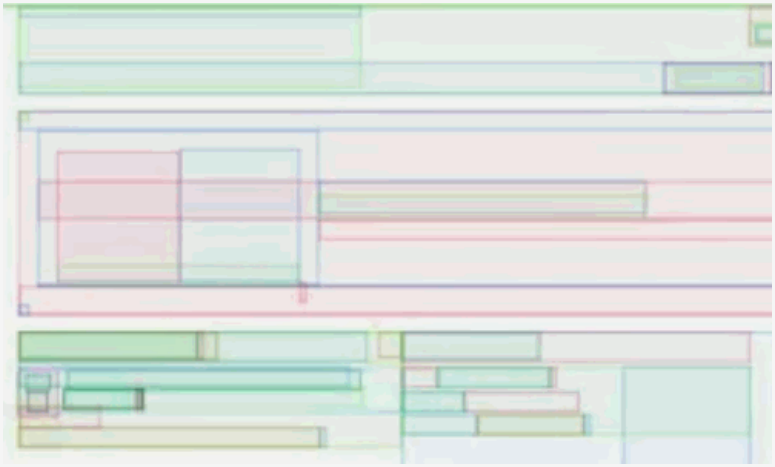
Les fichiers CSS et JavaScript ne sont pas les seuls à pouvoir êtres combinés pour réduire le no
généralement pour des icones ou de petites illustrations, la technique des sprites CSS est une b

Essayez de regrouper des images partageant la même palette de couleur. Si ce n'est pas le cas l
anormalement élevée car il sera sauvegardé au format PNG `true color` au lieu du classique P
encoder le résultat obtenu en base64 pour optimiser encore plus la taille de votre fichier.

- **base64** pour encoder vos images
- Un outil CSS : **SpriteMe**

ÉVITER LE CALCUL NAVIGATEUR AKA BROWSER REFLOW

Une fonctionnalité peu connue des développeurs en terme de performance est la construction d
navigateur. Des problèmes de performance apparaissent dès que l'on essaye de trop modifier d
force le navigateur à recalculer la place des objets tout en chargeant la page, ce qui ralenti le re



La chose la plus importante pour limiter le nombre de "reflow" est de limiter la taille du DOM et possible. Moins le DOM contient d'éléments, moins d'objets doivent êtres calculés et positionné et plus elle se chargera vite. Je ne le répéterai jamais assez :

> *JavaScript est pour le comportement, CSS pour le style.*

C'est pourquoi vous ne devez **JAMAIS** modifier vos éléments avec du JavaScript.

☐ Mauvaise solution

```
var el = document.getElementById('js-box');
el.style.color = "#fff";
el.style.backgroundColor = "#000";
el.style.borderColor = "#fc0";
```

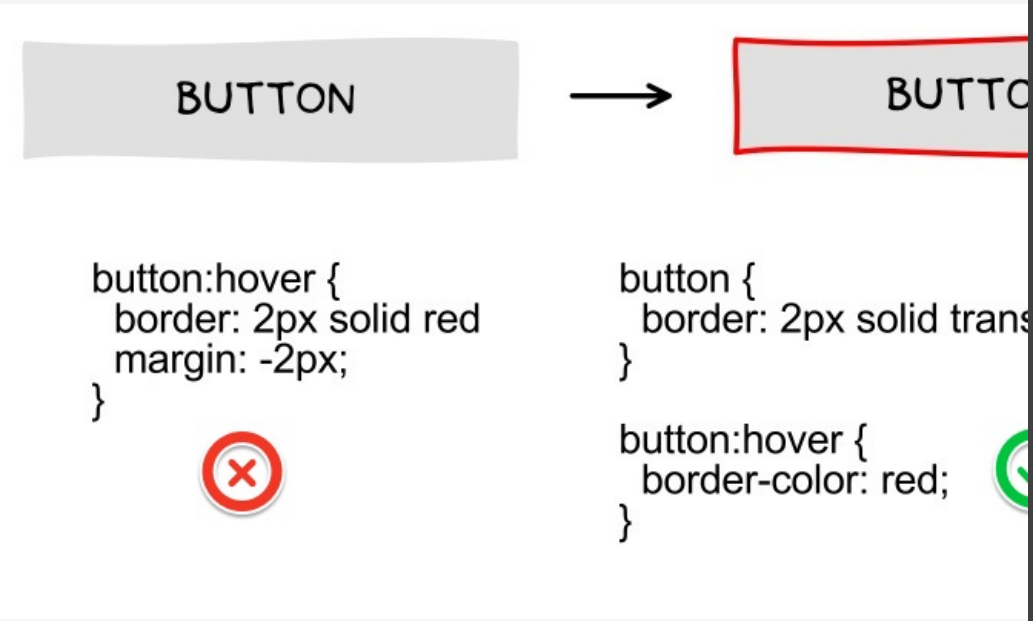
Chaque déclaration provoque un calcul. Dans notre cas il y a 3 reflow.

☐ Bonne solution

```
var el = document.getElementById('js-box');
el.className = "m-box";
```

1 seul reflow. le style est déclaré à sa place : dans une feuille de style et vous n'aurez pas besoi élément change.

Les calculs sont aussi impactés par l'ajout d'élements au DOM.



bon :hover et le mauvais :hover

Pour aller plus loin et comprendre le fonctionnement d'un navigateur web je vous invite à regar **Rouget développeur chez Mozilla**. Ou encore le projet **css-reflow-tracer**.

SUR MOBILE TOUCHER EST MEILLEUR QUE CLIQUER

L'impression de rapidité et la performance sont 2 choses différentes.

Connaissez-vous le célèbre délai des 300ms implémenté par tous les appareils mobiles ? Ce dél
temps d'effectuer un double-tap. Dans certains cas il n'est pas nécessaire et pour retirer ce déla

- **Comment créer des boutons rapides** ?
- Le polyfill **Fastclic.js**

ALLER À L'ESSENTIEL ET SUPPRIMER LES ÉNORMES FRAMEWO

JavaScript n'est pas le seul responsable dans l'explosion du poids des pages web, notamment d
bootstrapize. Le framework bootsrap de twitter est une excellente solution mais utilisez ses c
lui une solution plus légère comme Foundation ou encore KNACSS. Vous pouvez aussi envisage
avec **Suzy** et créer vos propres composants.

Il est facile d'inclure dans votre projet des librairies comme jQuery ou mootools, et d'y ajouter
développeurs nous devons nous poser la question de ce qui est vraiment utile. Si vous utiliser j
sélection `$('element')` regardez du coté de **Zest** ou **qwery**. Ces micro framework JavaScri
leur taille est d'environ 11ko comparé à plus de 90ko.

Comme expliqué précédemment, quand un navigateur télécharge jQuery, il doit parcourir le fic
bloque le rendu de la page. Soyez critique quand vous analysez vos besoins en JavaScript. 11k
seule chose dont vous avez besoin est `document.getElementByTagName('foo')`.

La performance comme budget

Généralement la performance ne fait fait l'objet d'un buget pour un projet web. Mais tout comm
ont une limite dans le nombre de polygones affichés à l'écran, les développeurs devraient impo
poids des sites internets.

Si votre client est résistant à se soucier de la performance, voici une petite astuce qui fonctionn

1. Sélectionner deux sites (au hasard le site web du client ainsi qu'un concurrent)
2. préchargez / precachez le site du concurrent en dur dans le cache (avec `localStorage`)
3. et l'exécuter en comparant le site internet du client non mise en cache.

Personne ne veut attendre et les gens sont impatient, à partir du moment où le client compren
web rapide, et investira pour avoir une bonne performance.

C'est culotté, mais cette méthode permet de démontrer l'importance de la performance. Je ne p
différente des publicicités télévisées que l'on voit à longeur de journée.

Pour aller plus loin, vous pouver créez un graphique comparatif du temps de chargement contr
verra la corrélation entre les deux. De nombreuses décisions devront être réexaminées dans le
une très, très bonne chose !

Publié par **David Leuliette**

☐ **@_flexbox** sur twitter

A **S'incrire au flux RSS**